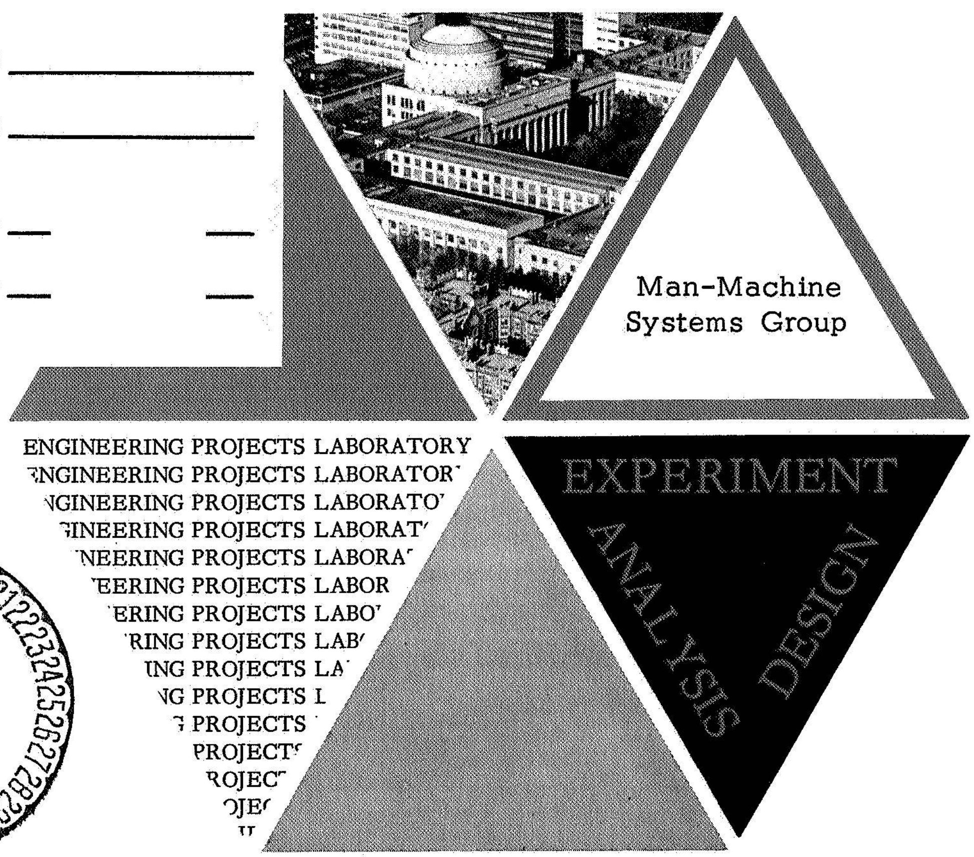
PROGRESS REPORT
April 1, 1967 - September 30, 1967
NASA Grant NsG 107-61

MEASUREMENT AND DISPLAY OF CONTROL INFORMATION
(Remote Manipulation and Manual Control)

Thomas B. Sheridan
William R. Ferrell

DSR 70283-6

Engineering Projects Laboratory
Department of Mechanical Engineering
Massachusetts Institute of Technology



DSR 70283-6

PROGRESS REPORT

April 1, 1967 - September 30, 1967

NASA Grant NsG 107-61

MEASUREMENT AND DISPLAY OF CONTROL INFORMATION

(Remote Manipulation and Manual Control)

by

Thomas B. Sheridan
William R. Ferrell

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Progress for this period is reported in two principle areas, Chapter 2 - Remote Manipulation and Chapter 3 - Continuous Manual Control.

In the Remote Manipulation area we are doing analysis and laboratory experiments in order to develop principles of human supervisory control of remote computer-manipulators. We seek to develop such systems, not only because of their promise and feasibility for performing a variety of operations in space without risk to human life or cost of life support equipment, but also because manipulation is a rich and well defined problem context for the study of those control functions the human is especially suited for vis-a-vis the computer.

Within the Remote Manipulation area we are presently concentrating on two kinds of problems: (1) the control interface between man and computer-manipulator; including considerations of command language or coding by which the human operator sets subgoals; (2) representation of the manipulation task within the computer; and the means by which the computer executes control actions to achieve the given subgoals.

Barber's "Mantran", a first computer language for human supervisory control is being exercised using the Man-Machine System Laboratory's AMF-8/PDP-8 computer-manipulator system. Barber is also studying the "fine structure" of manipulation control through time delay and command superstructure, and he is investigating the trading relation between loop transmission delay and process dynamic lags in relation to stability. Hardin is investigating the application of natural language syntatic analysis to the manipulation control problem.

Whitney is completing a study of the use of state-space techniques for representing manipulation tasks and finding optimal trajectories within these spaces to achieve stated goals. Harder is investigating the use of heuristics to achieve sub-optimal control trajectories for tasks too complex to make formal state space techniques practicable. Both Whitney's and Harder's work are expected to provide a better understanding of when a computer should assume control and when control should be turned over to a human operator.

Within the Continuous Manual Control area are three projects. D. C. Miller is continuing his research on the human operator as a bang-bang controller of second order systems; his present efforts are to predict deviations of human response from optimal response on the basis of component skills which together constitute the over-all control task and which are different for various displays used. R. A. Miller has completed a report on preview control systems which utilize one or two fast-time models; he has shown the relation of such systems to optimum control systems of the Weiner-Hopf type, and has fit his models to some experimental preview tracking data. Finally, Vickers is investigating preview control in a "maze" or discretized space where obstacles beyond the preview are unknown and the object is to minimize the path length to a goal state. Vickers's analysis may also be applied in the manipulation context.

2. REMOTE MANIPULATION

1. A Study of Fine Structure of Remote Manipulation - D. J. Barber

An experimental program has been started to investigate the micro-structure of human controlled remote manipulation. The hoped-for result is an understanding and classification of the invariants involved in this control task, with a view towards mechanization of some control-functions when manipulating through a time delay.

The experimental program consists of the following:

1. Commands were recorded for subjects performing a two-dimensional task, using another human (unseen) as the manipulator. Commands were given in English and the only feedback was from binary touch sensors on the "jaws" of the manipulator.

   Results:

   a. All motion commands can be classified as belonging to five motion routines; search, grasp, carry, put, and error adjustment.

   b. The number of commands increases with increasing ambiguity in feedback. This ambiguity depends on the number and size of the touch sensors, and on the shape and size of the object being grasped.

   c. The size of commanded motions decreases with increasing risk. Risk can be related to the humans estimation of the probability of making an error.

2. A program has been written to provide a variable time delay for two analog voltage signals. This program will be used with a two-dimensional, continuous control task to investigate the effect of time delay on control of a plant with dynamic lags. The hypothesis is that small time delays will become relatively unimportant in the control of plants with large lags, and that continuous control will be stable, eliminating the need for a move and wait strategy.

3. Programs have been written to provide several different control systems for the AMF-8 seven degree of freedom manipulator. These programs provide for positional master-slave control, proportional rate control, on-off rate control with a shoulder mounted joystick, and control of rate or position with a model. These programs all incorporate a variable time delay. A compiler language which accepts typewritten commands in English is also available.

A manipulation task is now being constructed and data will be recorded for operators performing the task with all of the control systems mentioned above. These data will be examined for motions common to all control systems, and a relationship will be sought between performance measures and parameters of the control systems.

2. Manipulator Control Through Natural Language - P. A. Hardin

The purpose of this project is to develop a system which "Understands" typed imperative statements or commands, composed from a restricted set of verbs, nouns, adjectives, adverbs and prepositions, and appropriate to dynamic control of a manipulator. In the process of this development we hope to better understand the heirarchical nature of manipulating physical objects in space, and the analogies which seem to exist between selecting and ordering hand movements and selecting and ordering words to control physical actions.

The system is being designed to consist of a cascade of three processes:

1. A sentence parser, which recognizes the typed input words and casts them into a structure relating the words for named objects, goals, actions to be taken, etc. (i.e., "verb" an "object" to a "place" in an "adverb" fashion).

2. A semantic interpreter, which operates on the structured statement so that it can "understand", i.e., decide on unique subgoals. This is not to infer that the computer will always understand the same subgoals the human meant it to understand.

3. A manipulation interpreter, which, given the understood subgoal, decides upon a sequence of primitive manipulator actions to achieve that subgoal. This third process may make use of state-space algorithms, such as are described in this report by Whitney, or may use heuristic techniques as described by Harder.

It is presumed that very little information will have to be carried between the three processes, though feedback from each will be programmed to affect the prior one, and may also be communicated to the human. For example, at the parsing level, the operator may be informed that he is using illegal words or grammar, at the semantic level may be told that his (grammatical) sentence makes no sense for manipulation ("close the jaws left"), at the manipulation interpreter level he may be told that there is a remembered object in the way or that some limit ought to be set on how far the jaws should move if contact is not made.

Starting with simple manipulation tasks in two dimensions (on a scope face) and a very restricted word set, programs are being written for an experimental simulation.

3. <u>State Space Models of Remote Manipulation Tasks</u> - D. B. Whitney

The following abstracts Whitney's recently completed PhD thesis by the same title:

Remote manipulation is difficult enough if the operator is close to his work, because he is never close enough, the feedback is often meager, and the apparatus is frequently clumsy and hard to control. Add to this a significant time delay and efficient manipulation becomes almost impossible. Our goal is to equip the manipulator with some intelligence of its own so that we may give it fairly general commands oriented towards goals at the operator's level of concern, and it will be able to translate these into commands which are oriented toward goals at the motor level, commands directly intelligible to the manipulator's prime movers. The computer should also be able to interpret local feedback sent to it by touch sensors on the manipulator during execution of these commands. This would relieve the operator of the petty details concerning motion or touch whose delayed transmission or receipt causes so many difficulties.

The approach is to consider the manipulator's hand and the task site as a system to be controlled by the operator. (This differs from the usual approach in which only the manipulator's hand is included in the system model). This system is dealt with from the point of view of Modern Control Theory: we wish to transform the system from the current state (configuration of objects and hand) to another, desired state. All commands are therefore abstracted to the form "New desired state is..."

To implement this approach, the system must be thought of as having a state vector description and an equation of motion. The latter relates the allowed transitions of the system's state to the allowed set of motor level commands. These commands are limited to a few in number and are static and incremental. For example: "Move the hand one inch left and stop." (If several of these are executed in quick sequence, the intermediate stops are ignored and one continuous motion results.) However, since the system state vector contains locations of objects, one may include commands like "Carry object A one inch left" or "Push object A one inch left."

The computer, upon being told to pick up and move a certain object to a certain location without bumping into any other objects, must find a sequence of these elementary commands which will do the job. This sequence corresponds, by way of the equation of motion, to a path in the state space. This space describes all configurations of the hand and environment, and is generated in the computer by successively considering the application of each allowed motor level command to each configuration. The result is a map of all possible ways in which all possible tasks can be carried out in the given environment using the given set of motor level commands. Since many of these ways overlap somewhat and accomplish the same task with widely varying degrees of efficiency and directness, an optimality criterion is used to choose the way which is most efficient or desirable in some sense. The resulting path, being a sequency of elementary commands corresponding to prestored routines, reads like an ordered work description to the manipulator's hand, and constitutes a plan for executing the operator's desire.

The hand presumably is equipped with touch and force sensors. Information from these sensors is to be evaluated for the most part by the computer. These sensors tell the computer when the hand has collided with an object, or how big an object is, or how tight the hand is grasping some object. The computer, executing the path, knows what sense inputs to expect if the plan goes smoothly. Unexpected obstacles may be incorporated into the state space upon discovery. The computer, knowing how far along the aborted plan has gone, can generate a new one using the new information and the same methods as before. Only in case of difficulty will digests of such sensor data be transmitted to the distant operator.

The state space idea, combined with several heuristics, has been demonstrated on a simple plotting table manipulator. Objects may be referred to by name, picked up, carried to named locations, and so on. The computer keeps track of all such changes in the environment. Only the physical size of the task site limits the number of objects which may be kept track of in this way.

A doctoral thesis entitled State Space Models of Remote Manipulation Tasks by D. E. Whitney, is being published as M.I.T. Engineering Projects Laboratory Report No. 70283-5.

4. Heuristic Control of Remote Manipulation - B. M. Harder

This is an effort to define both what a computer can do by itself and when the operator must intervene to substitute a subgoal or series of subgoals in place of a subgoal too complex for the computer to handle by itself. The project is viewed as a complement to Whitney's thesis, which deals purely with algorithms to determine trajectories in state space - a powerful approach, but one requiring far too much computer memory for tasks of any complexity.

Assumptions are that the computer will keep a running list of all obstacles and construct motions to avoid them. Suboptimum paths will be acceptable. The computer will be programmed with a set of heuristics which will be "tried" according to yet unspecified priority criteria. The operator will communicate goals, priorities and other criteria symbolically.

Initial effort will concentrate on the task of finding an acceptably short path from given starting point to given goal through a two-dimensional obstacle field, where positions and shapes of obstacles are known. Heuristically, a set of straight line segments tangent to the obstacles' peripheries will be evaluated and a path chosen from these (Fig. 1). This requires evaluation of a far smaller set of paths than a state space approach would require. Under certain conditions the program will call upon the human operator (1) to select among those obstacles worth considering (eliminating paths around obstacles far off any sensible course) or (2) to specify an easier subgoal.

A taxonomy of manipulation tasks will be attempted, in terms of which different heuristics will be constructed. It is expected that initially the size of this set of tasks or subtasks (and corresponding heuristic routines) will be about ten, and will subsume, hopefully, most two-dimensional manipulation tasks which one might consider.

One example of a category of tasks different from finding a shortest path around obstacles is illustrated in Fig. 2. The object will fit through a "gate" by itself but the manipulator jaws will not. The object must than be inserted from one side, released, then the jaws must be carried (along some other available path or a second set of jaws brought into play) to a position from which the object may be regrasped and pulled through the hole. Threading a needle and placing a bolt in a hole fit this category.

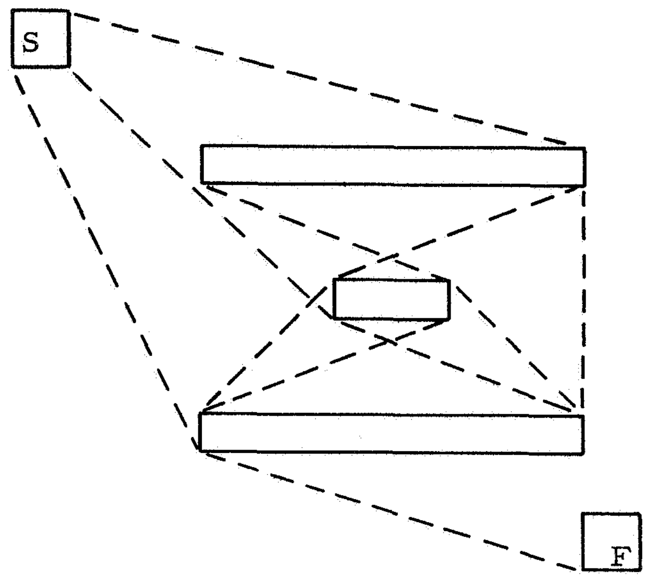


Fig. 1

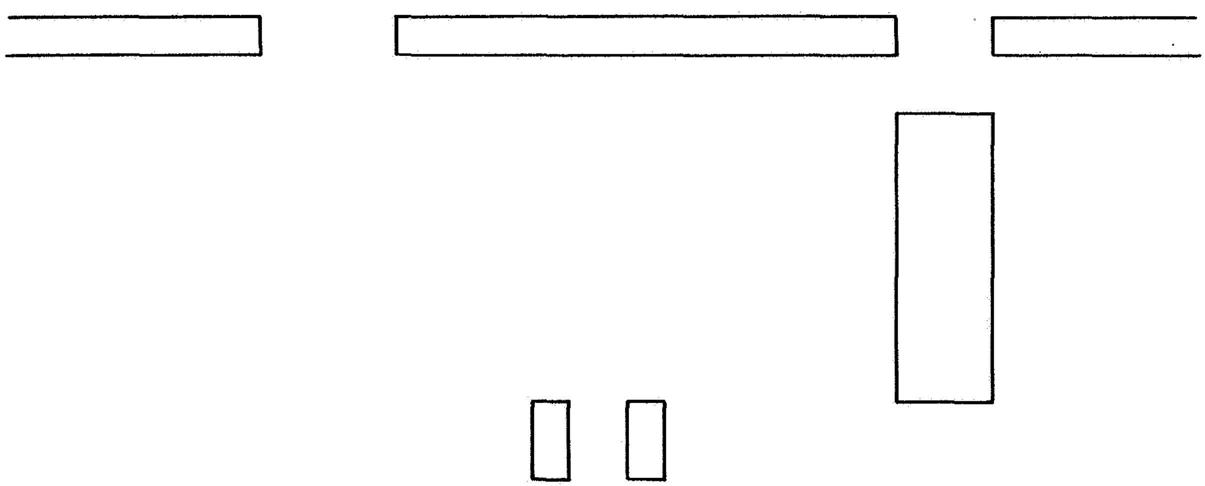


Fig. 2

3. CONTINUOUS MANUAL CONTROL

1. The Optimality of the Human Controller as a Time-Optimal Bang-Bang State Regulator of Second Order Systems - D.C. Miller

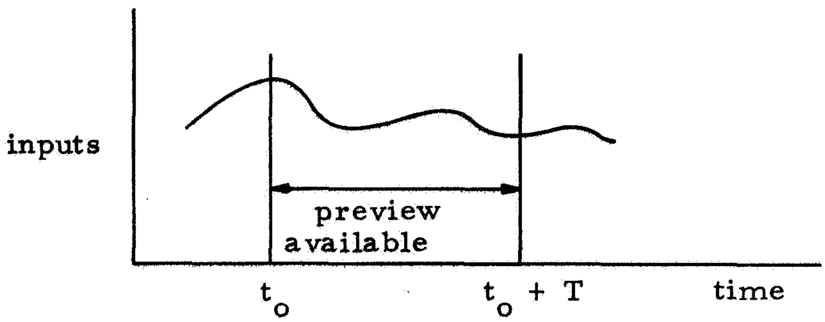
D. Miller has compiled data on the switching performances of three subjects for each of four second order systems using each of four types of displays. The level of performance, as measured by the total time taken to bring the system state to zero from a series of initial states, has been shown to depend significantly on the nature of the system and the nature of the display. There proved to be no statistically significant difference between subjects, and no significant interaction between the system and the type of display.

Studies are now underway the goal of which is to account for the differences in performance in terms of the basic psychomotor skills required. Hopefully, a theory can be developed which will predict a subjects' performance from the parameters of the system and the display being used. One component skill already investigated is that of switching at the instant that a moving point crosses a line. It appears that the standard deviation of the switching error (in terms of the distance from the actual intersection point) is proportional to the speed of the moving point. This indicates that the subjects had a constant standard deviation in time at which they switched. This standard deviation was approximately 50 ms. for all three subjects used.

This result will be used to predict the distribution of switch points which occurs with an actual system and a display in which the switch line is explicitly indicated. If this prediction is successful, further tests will be made of the additional psychomotor skills necessary with other displays in which the switch curve is not shown.

2. Preview Control Systems with One and Two Fast-Time Models - R.A. Miller

Assume an operator is attempting to control a given plant such that it follows a given input index. Also, assume that he has a preview of the future input to the system for a finite distance. Essentially we have the following (Fig. 3).



Plant: $\dot{x} = f(x, u, t)$

x: state

u: control

t: time

$$\text{Performance Index } J = \int_{t_o}^{\infty} L\,(x, u, t)\,dt$$

Fig. 3

It is not possible to solve the problem on an infinite interval unless the input is known on the entire interval. A possible method of solution involves solving for the optimal control over the known closed interval (t_o, T), then travel a short distance δ and recompute a new control over the new interval $(t_o + \delta, t_o + \delta + T)$. The net result is a good approximation to the optimal trajectory over the infinite interval.

Over any given closed interval, the problem is easily solved using any of the methods of optimal control. The thesis by R.A. Miller[+] followed this procedure for linear systems and quadratic performance indices. The result is an equation for the optimal control based on <u>future</u> error over the interval of interest $(t_o, t_o + T)$.

This implies that the operator, if he is to optimize the performance index, will generate an estimate of the future response and weight the estimate response by the terms of the control equation to determine the control he should use.

This procedure is consistent with several previously proposed models of the human operator in such situations. Specifically, two-time scale modeling provides a direct method of implementing the solution of the optimal control problem.

As originally proposed, a dynamic mode of the controlled plant was used in a feedback loop, operating faster than real time, to provide an estimate of future error. The Miller thesis shows this system is not satisfactory. The fast-time model must duplicate a system with preview--which the feedback model does not have. This leads to cascading of fast-time models to provide preview in the response estimation. This procedure is essentially iterative, with successive iterations, i.e., repeated levels of models, providing better estimates of the resulting response.

It was shown that an optimum preview time T exists for any given system, depending on the plant, performance, index and accuracy of prediction of future response. That is, $\int_{t_o}^{\infty} L(x_1, u, t)dt$ is minimized when a specific value of T is used for the successive short interval problems. The magnitude of T is a trade-off between accuracy and plant dynamics. Accuracy of the response estimate decreases with increasing preview time, but if preview is cut too short significant information is lost, especially for sluggish plants. With high gain, i.e., fast responding plants, the preview must be shortened to preserve accuracy. It was also shown that preview, even when of poor quality, significantly improves performance. The greatest improvement comes with marginally stable, difficult to control plants.

It appears that the modeling procedure can easily be extended to situations involving noisy inputs. There also appears to be a connection between this type model and information type models.

An additional result may be significant. The computational techniques, i.e., iterative fast-time models, might be useful for solving optimal problems with analog computers.

3. A Goal-Directed Maze Solver - W. H. Vickers

Introduction: The problem being considered is how to find a path through a maze to a known end state given only local information about which transitions are possible. In other words, assume you can see the possibilities open to you for a few steps down the road and you know the general direction in which you want to go, but you do not know what obstacles or dead ends you may run up against beyond your visual preview. An example may help clarify the problem (Fig. 4).

A preview of 1 unit would allow seeing A and B. A preview of two units would allow seeing, A, B, C, D, and E BEFORE taking the first step to either A or B. A preview of 3 units would allow seeing A, B, C, D, E, F, G, H, and J. After taking the first step to B then the whole process is repeated.

From here a preview of 2 units allows you to see S, A, D, E, G, H, and J before you take your second step to D, E or (possibly but not likely) S. (Fig. 5)

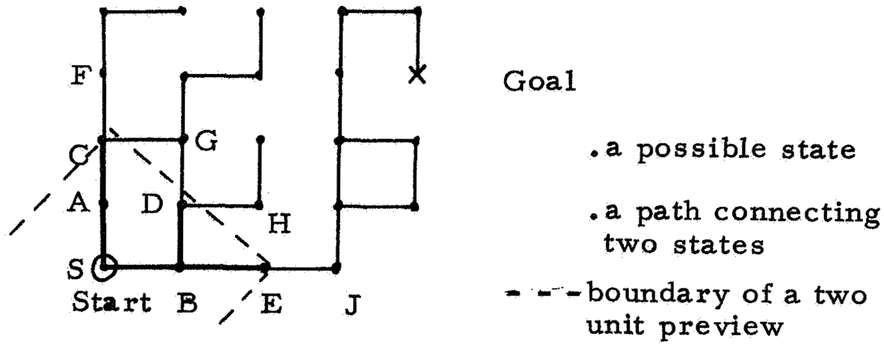

F
C
A
D
G
H
S
Start
B
E
J
Goal
. a possible state
. a path connecting two states
- - - boundary of a two unit preview

Fig. 4

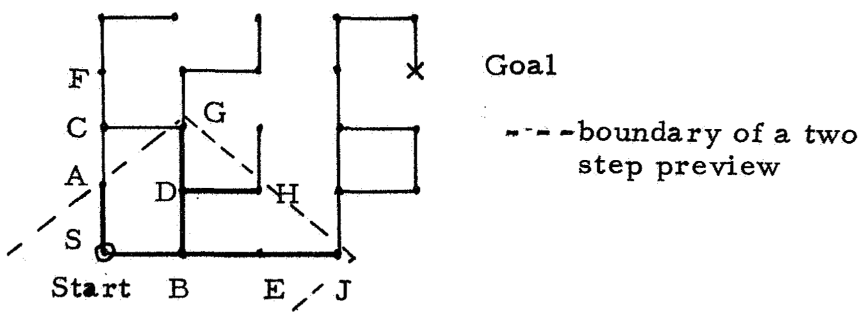

F
C
A
D
G
H
S
Start
B
E
J
Goal
- - - boundary of a two step preview

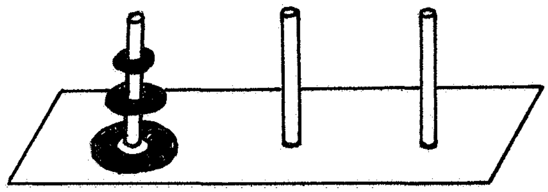
Fig. 5

The maze problem as formulated here was conceived as a model of people and grew out of Sheridan's idea of preview control. Instead of a one-dimensional control task where the operator must try to control a continuous dynamic system to follow a signal, which is the case the preview control problem considered previously, this case is where the operator has no signal to follow but must make his own nominal path. This is true in many tasks performed by people, especially obstacle avoidance problems.

For example the maze solver could be thought of as a model of a blind man with a cane. Other examples include maneuvering a submarine through a mine field with limited sonar, or flying an airplane beneath the enemy's radar in hilly country where one might fly around a mountain rather than over it, or even fly a small plane on a long flight with the possibility of small storms that could be avoided. Other problems include controlling a remote vehicle or manipulator on the moon where local craters and rocks are unknown until they come within range of the manipulator's camera. All of these examples involve spatial movement and this is in fact the condition that is most vividly modeled by the limited preview idea. However there are some purely thought problems that people solve which, with some imagination, can be modeled as a limited preview, goal-directed maze solver. For example single person games such as the tower of Hanoi problem (Fig. 6 and 7) have a maze structure similar to the one described above and have a known goal. The way people usually solve it is to imagine what would happen if they move a certain piece and continue to imagine until they are several plies down the maze before they take the first move. This brings in the limited preview, although of course it is not constant in all directions of the maze, nor is it constant from one move to the next.

The difference between the maze-solver and these mental exercises such as games or theorem proving is basically that the maze solver has a well defined metric for measuring the distance to the goal which these harder problems do not.

In addition to spatial problems which involve a real preview limitation it is conceivable that there will be spatial problems which, like many games,

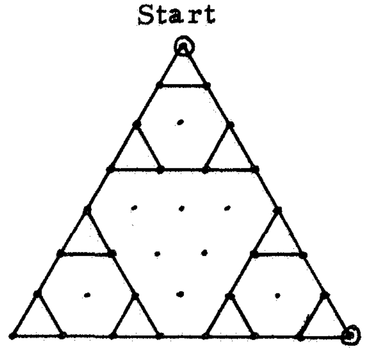


physical layout

get the rings to post No. 2
only smaller rings can sit on top of a ring

Fig. 6

Start



Finish

maxe for problem

Fig. 7

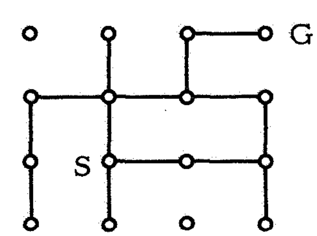
have such a tremendous number of possible states that shortest path problems cannot be solved by any of the algorithmic approaches in a reasonable time. For example in a manipulation problem the state space grows very fast with the number of degrees of freedom and the number of objects to be moved (see discussion in this report by Whitney).

Programming Progress - A program was written as a first attempt at a model of a maze solver with a fixed preview of only one unit in all directions. The program is also able to "see" the number of new moves from a test state although it is not allowed to see in which direction (s) they go (See Fig. 8 and 9).

The program does not use this information to reconstruct the maze or to make hypotheses about the maze. Rather this information is used to evaluate a cost function associated with each of the reachable test states. The cost was taken as a linear combination of: (1) direct distance from the test state to the goal (d in Fig. 3) and (2) the number of new moves from the test state, and (3) a bias for having been to the test state before (a record being kept of all past states it has actually been to). The constants are adjusted so that the model, always picks an unexplored state over one it has already visited, and it will pick the state with more new moves if the other states do not have much distance advantage.

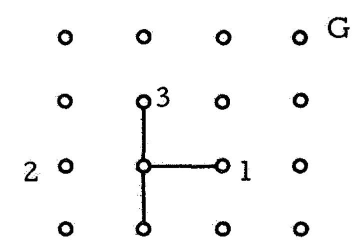
Whenever all reachable moves have already been explored, it recognizes the situation as a dead end. In this case it in effect backs up until it gets to the state that has a new move. There are at least two ways to accomplish this back-up, one of which is faster but can lead to disasterous results while the other can be impossibly slow. A better way might be to remember the last good move passed and then go back to it any way possible. This procedure could also be used in case one area seems fruitless. If all moves available were sufficiently worse than the move remembered the procedure could go back, even though the current direction is not really a dead end.

Probability Considerations - The maze problem is such a well defined problem as heuristic problems go, with such a nice metric, that it would seem possible to use probability theory to come up with a rational criterion for choosing which way to go. To do this, some assumption must be made about



actual maze

Fig. 8



what the program can "see".
numbers refer to number of new moves from their state.

Fig. 9

the maze. The simplest assumption one can make is to assume that each one-step link is chosen independently with the same probability p. This assumption of course neglects any correlation that the maze may have.

A good first step in finding a rational criterion seems to be to find the expected shortest path from a point to the goal. If this were known then we could objectively compare two states.

Some progress has been made in calculating expected shortest path lengths between two points. I hope to get this calculation automated. Also I am working on the analytical calculation of expected path length from several connected points to a common goal.

Application to Manipulation - The limited preview approach could prove very useful in computer controlled manipulation since the potential memory reduction is substantial. Also the local maze information would not really have to be stored since the information is available through the sensors all the time. In other words, we can use the real situation as the computer's memory.

Therefore, in an effort to see how readily this maze solving technique adapts to manipulation problems, I worked through a simple example without using the computer. The physical situation is in Fig. 10.

The task is to get the jaws to location (3, 2). The jaws do not rotate and they are either opened or closed. The state of the system is represented by a 5-dimensional vector space (X_{obj}, Y_{obj}, X_{jaw}, Y_{jaw}, ω_j) where ω_o = width of jaws = open or closed.

The rules for proceeding through the task are as follows:

Looking only one step ahead take the move which reduces the geometric distance to go in the first four coordinates by the largest amount. If all are equal (or all make it worse) reduce the distance in the 5th dimension. Do not include states that have already been visited in this list of possible states.

If you get to a point where all available moves have been visited, back up toward the last available move. The goal is (ϕ, ϕ , 3, 2, c) where ϕ repre-

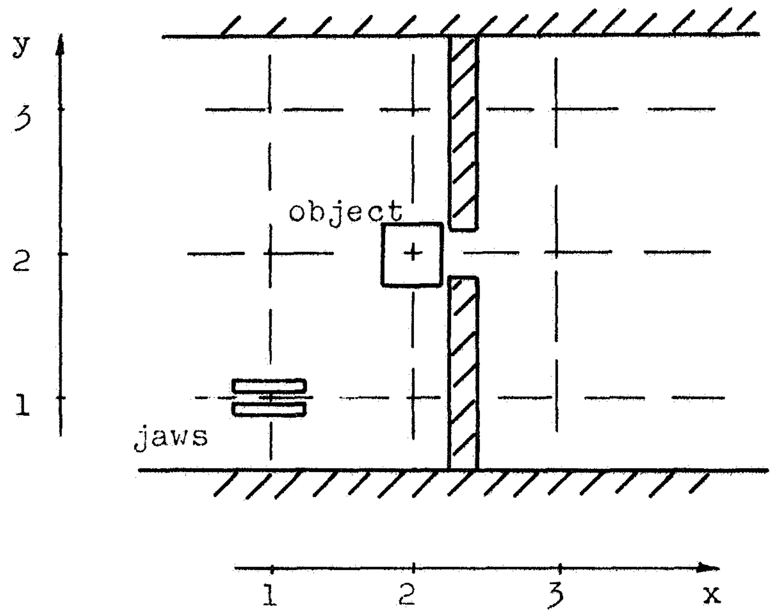


Fig. 10

sents "don't care". The 5-dimensional maze structure is shown in Fig. 11 in four, 3-dimensional graphs (two other graphs are omitted since they show nothing of interest). The progress using these rules is as follows: (underlined numerals refer to dimensions which can be changed to get to an unvisited state).

Start

(5) $(2, 2, \underline{1}, \underline{1}, \underline{c})$ Goal: $(\phi, \phi, 3, 2, c)$

(1) $(2, 2, 2, 1, \underline{c})$ This was essentially a dead end

(2) $(2, 2, \underline{2}, 1, 0)$

(3) $(2, 2, 1, \underline{1}, 0)$

(4) $(2, 2, \underline{1}, \underline{2}, \underline{0})$

(5) $(2, 2, 2, 2, \underline{0})$ ----- It has got the object surrounded

(6) $(\underline{2}, \underline{2}, \underline{2}, \underline{2}, c)$

(7) $(2, 1, 2, 1, \underline{c})$ ----- Has to back up and does

(8) $(2, 1, \underline{2}, 1, 0)$

(9) $(2, 1, 1, \underline{1}, \underline{0})$

(10) $(2, 1, \underline{1}, \underline{2}, \underline{0})$

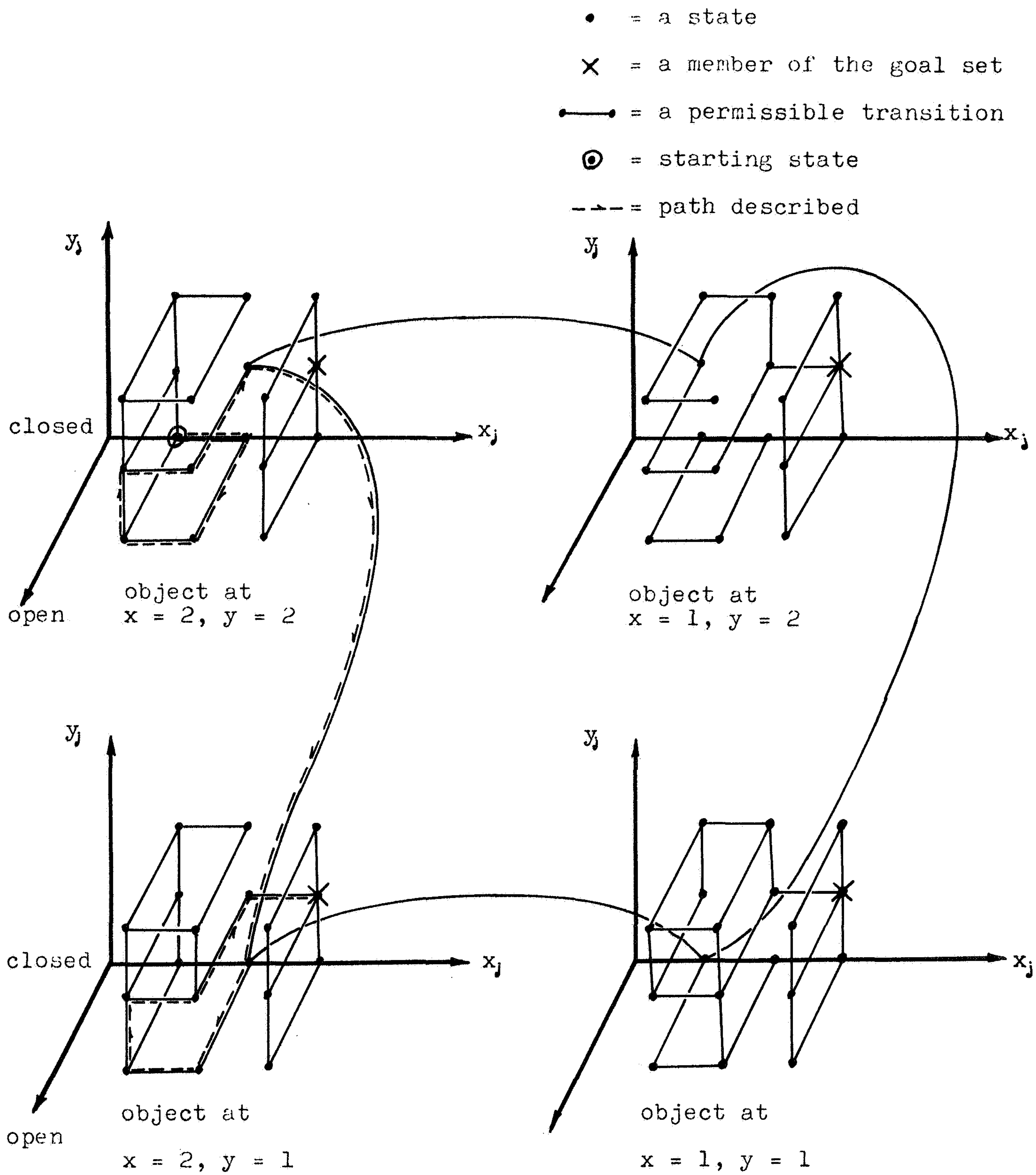
(11) $(2, 1, 2, \underline{2}, \underline{0})$ -----Good choice

(12) $(2, 1, \underline{2}, \underline{2}, c)$

(13) $(2, 1, 3, 2, c)$ -----Goal or a member of the goal set.

There are several points that this example brings out. First, the process of putting ϕ (don't care) into those coordinates not specified in the Goal and then calculating the distance to go without considering these coordinates means that we are trying to get to that goal state which is closest to our current position. That is, up through steps 6 we are going toward Goal No. 1 $(2, 2, 3, 2, c)$ whereas after that we are going toward Goal No. 2 $(2, 1, 3, 2, c)$. This can be represented schematically as in Fig. 12.

The second point raised by this example is the concept of a metric. It was nice to work in Euclidean space but already we are out of it. We have to ask if all coordinates are weighted equally in determining distance and we decide that they are not. The jaw opening and closing is not as important as position. What I have done is something like saying distance



= a state
= a member of the goal set
= a permissible transition
= starting state
= path described
y_j
x_j
closed
open
object at
x = 2, y = 2
object at
x = 1, y = 2
object at
x = 2, y = 1
object at
x = 1, y = 1

Fig. 11

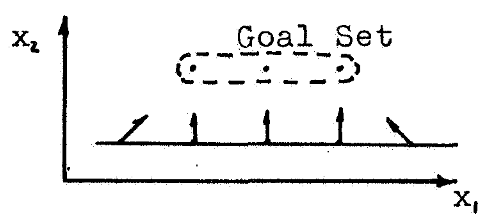


Fig. 12

from state a to state b, $\Delta_{ab} = \left[\Delta a^T{}_{ab}\, G\, \Delta x_{ab}\right]^{1/2}$ where Δx_{ab} is the difference, between state vectors associated with a and b, and where G is a diagonal matrix. In the case of this example.

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \epsilon \end{pmatrix} \quad \text{with } \epsilon \sim 0.1$$

One other minor point: in manipulation we must allow for diagonal moves, when we are carrying an object for instance. This does not affect the basic idea but might change some calculation of expected shortest path.

The fact that this limited preview concept can handle manipulation type problems is encouraging. It took 13 steps to complete a task that only required 9 if done optimally, but this is a minor point.